EQUALITY-SUPPORT COVERS FOR EXACT BRANCH-SET PRESERVATION UNDER FIXED ACTUATOR INTERFACES

PHILLIP KINGSTON

This work is licensed under a Creative Commons Attribution-Noncommercial-ShareAlike 4.0 International License
(CC BY-NC-SA 4.0)

Abstract. A fixed actuator interface can preserve safety by pruning viable branches yet fail to reproduce the target branch set exactly. We study the minimum number of predeclared actions used by a freely designed memoryless state command on a target-closed finite domain. For total deterministic global event-disable masks with viable uncontrollable transitions, exactness requires disabling every hazardous controllable event while retaining an enabled event in each viable successor fiber. Successor-injective effects collapse exact supports to disjoint hazard-set classes and make exact minimization polynomial, whereas safe supports remain overlapping containment sets and safe minimization is NP-hard. Duplicate effects can realize any finite set system simultaneously as the safe- and exact-support system. Pointwise nested interfaces instead yield exact-action intervals and a polynomial interval-transversal problem. Both restricted families attain the maximal separation among instances with finite q_{Exact}: $q_{\mathsf{Safe}} = 1$ and $q_{\mathsf{Exact}} = |K|$. A fixed-interface factorization connects ideal history-dependent pruning to these finite equality-support covers. One motivating instance is tool gating for autonomous agents, whose permission profiles are global event-disable masks of exactly this kind.

1. Introduction

A nondeterministic process exposes a set of possible futures rather than a single future. A restrictive runtime mediator can make a safety condition hold by discarding enough futures, possibly retaining only one conservative branch. The stronger requirement studied here is *exact branch-set preservation*: the mediated state-run set must equal the target state-run set, so every target-extendible successor is retained and every other successor is rejected. The equality is over unlabelled state runs: distinct events with the same successor are intentionally identified, so the objective is neither event-labelled trace equality nor bisimulation. It is more demanding than safety-only pruning and distinct from trace-level notions of transparency used in runtime enforcement and shielding (Schneider, 2000; Ligatti et al., 2005; Bloem et al., 2015; Hsu et al., 2024).

Date: 22 July 2026.
Key words and phrases. exact branch-set preservation, fixed actuator interface, equality-support cover, global event-disable mask, event-effect fiber, successor injectivity, interval transversal, Set Cover, viability kernel, supervisor reduction, tool gating, agentic artificial intelligence.
Email: phillip.kingston@gmail.com. ORCID: 0009-0005-2256-3079.

The central question is whether a fixed actuator library can reproduce the ideal target filter, and how many library actions are needed when one may freely design a memoryless state-to-action map. A representative setting is a platform exposing a fixed catalogue of global firmware interlock modes. Each mode disables the same named commands wherever they occur, while uncontrollable transitions remain available. The controller may select different modes at different states but cannot synthesize a new mode by combining state-specific disable decisions. The selected-action count can then represent the number of modes that a deployment must enable and certify.

A second instance arises in tool gating for AI agents. An agent's controllable events are its tool invocations; environment responses are uncontrollable. A platform typically ships a fixed catalogue of permission profiles, each disabling the same named tools wherever the agent's task stands, rather than synthesizing a per-state policy. Such a profile is a global event-disable mask in the sense of Section 5.2, and the cover numbers price how many profiles a deployment must maintain to run safely versus to preserve exactly the intended task behaviours. All cover numbers remain relative to the declared library and implementation domain; they do not measure sensing, state estimation, computation, communication, or physical actuator design.

The main contribution is to identify how restricted actuator interfaces shape the resulting support system. For total deterministic global event-disable masks, let $H_D(x)$ be the controllable events that leave the viability kernel at state x, and let $\Phi_x(y)$ be the events with successor y. Theorem 5.16 characterizes exactness by $H_D(x) \subseteq B$ together with survival of at least one enabled event in every viable effect fiber. Under successor-injective effects, Corollary 5.18 reduces this condition to $B = H_D(x)$, so exact supports are disjoint hazard classes and the feasible optimum is the number of distinct represented classes.

Successor-injective effects do not similarly simplify safety. Safe supports retain the containment form $H_D(x) \subseteq B$, and Proposition 5.20 realizes arbitrary Set Cover instances, up to padding states common to every support. Thus exact minimization is polynomial in this mask regime, whereas the safe threshold problem remains NP-complete. Theorem 5.19 gives the complementary noninjective result: duplicate effects permit any finite set system to be realized simultaneously as the safe- and exact-support system in a restricted self-loop-or-sink construction; the corresponding hardness results then follow by Set Cover.

Once supports are formed, minimum repertoire selection is ordinary Set Cover; the structural results identify when restricted interfaces force intervals or disjoint classes and when duplicate effects retain arbitrary geometry. Pointwise nested interfaces yield exact-action intervals and reduce exact minimization to the classical interval-transversal problem (Theorem 5.14), while nested and successor-injective mask families attain the maximal separation among instances with finite q_{Exact} (Theorem 5.25). The fixed-interface factorization connects the ideal filter to these support covers. The objective differs from state-partition-based supervisory control on the original plant state space (Butez et al., 2014), from supervisor reduction and control covers (Vaz and Wonham, 1986; Su and Wonham, 2004, 2018), and from invariant-cover cardinality: here E_D is fixed first, each action is a predeclared global postset map, and the objective counts distinct maps selected on plant states rather than supervisor states or observation cells.

The paper is organized as follows. Sections 2–4 give the ideal pruning baseline and fixed-interface factorization criterion. Section 5 introduces finite viability and support covers,

and its subsections establish the structured interface results, the event-effect classification, and the comparison theorems. Section 6 positions the contribution and states the model limits, and Section 7 concludes.

2. Runs and extensional specifications

Let $P = (X, T)$ be a serial transition system, with X discrete and $T \subseteq X \times X$. Fix $x_0 \in X$ and write

$$\Omega = \mathrm{Runs}_P(x_0) = \{\tau \in X^\omega : \tau_0 = x_0,\ (\tau_t, \tau_{t+1}) \in T\ \forall t\}.$$

Let $\mathrm{Hist}_P(x_0)$ be the finite plant paths from x_0. For $h = x_0 \cdots x_n \in \mathrm{Hist}_P(x_0)$, let $\mathrm{last}(h) = x_n$, let $\mathrm{Post}(h) = \{x : (x_n, x) \in T\}$, and let $[h]_\Omega$ be the relative cylinder of runs extending h. These cylinders generate the relative prefix topology on Ω. An extensional specification is a set $A \subseteq \Omega$.

3. The ideal exact-realization baseline

Definition 3.1 (Nonblocking successor-pruning enforcer)**.** A nonblocking successor-pruning enforcer is a total map assigning a nonempty set $g(h) \subseteq \mathrm{Post}(h)$ to every $h \in \mathrm{Hist}_P(x_0)$. Write $\mathrm{Runs}(P, g)$ for the infinite runs that always use retained successors.

For $A \subseteq \Omega$, write $\mathrm{Pref}(A)$ for its finite prefixes and define the ideal target filter

$$F_A(h) = \{x \in \mathrm{Post}(h) : hx \in \mathrm{Pref}(A)\}, \qquad h \in \mathrm{Pref}(A). \tag{1}$$

The following standard facts isolate the semantic target later factored through a fixed interface. Safety properties are precisely the closed sets in the relative prefix topology (Alpern and Schneider, 1985, 1987).

Lemma 3.2 (Exactness determines the local filter)**.** *If* $\mathrm{Runs}(P, g) = A$*, then* $g(h) = F_A(h)$ *for every* $h \in \mathrm{Pref}(A)$*.*

Proof. Every target prefix is reachable under g: if $h \in \mathrm{Pref}(A)$, some run in $A = \mathrm{Runs}(P, g)$ extends h, so h is a controlled path. A target-extendible successor must therefore be retained. If a retained successor were not target extendible, totality and nonempty retained postsets would provide an infinite controlled continuation outside A. □

Proposition 3.3 (Closed-target realization under ideal pruning)**.** *A set* $A \subseteq \Omega$ *is the exact outcome set of a nonblocking successor-pruning enforcer if and only if* A *is nonempty and closed in the relative prefix topology.*

Proof. For nonempty closed A, retain $F_A(h)$ on $\mathrm{Pref}(A)$ and choose any nonempty postset elsewhere. The outcomes are the infinite branches through $\mathrm{Pref}(A)$, namely A. Conversely, the complement of any enforcer outcome set is a union of cylinders rooted at rejected steps, hence open; nonblockingness gives a nonempty outcome. □

This baseline assumes that every nonempty successor subset is an available intervention. The remaining sections replace that assumption by a declared finite action alphabet.

4. Factoring exact enforcement through a fixed action interface

Let $A \subseteq \Omega$ be nonempty and closed.

Definition 4.1 (Finite-alphabet history interface)**.** A finite-alphabet history interface consists of a nonempty finite action set U and nonempty postsets

$$\varnothing \neq \mathrm{Post}_u(h) \subseteq \mathrm{Post}(h)$$

for every $h \in \mathrm{Hist}_P(x_0)$ and $u \in U$. A finite summary is a total map $\eta : \mathrm{Hist}_P(x_0) \to Z$ to a nonempty finite set Z, and a selector is a map $\mu : Z \to U$. They induce the total enforcer

$$g_{\eta,\mu}(h) = \mathrm{Post}_{\mu(\eta(h))}(h).$$

Finiteness of U and Z alone does not imply that the history maps are computable or finite-state; the definition is extensional.

For $h \in \mathrm{Pref}(A)$ define

$$U_A^{=}(h) = \{u \in U : \mathrm{Post}_u(h) = F_A(h)\}, \qquad U_A^{\subseteq}(h) = \{u \in U : \mathrm{Post}_u(h) \subseteq F_A(h)\}.$$

Theorem 4.2 (Fixed-interface exactness criterion)**.** *Let A and the finite-alphabet history interface be as above.*

(i) (η, μ) realizes A exactly if and only if

$$\mathrm{Post}_{\mu(\eta(h))}(h) = F_A(h) \qquad (h \in \mathrm{Pref}(A)). \tag{H=}$$

(ii) For a fixed summary η, an exact selector exists if and only if

$$\bigcap_{h \in \eta^{-1}(z) \cap \mathrm{Pref}(A)} U_A^{=}(h) \neq \varnothing \qquad (z \in \eta(\mathrm{Pref}(A))). \tag{F=}$$

(iii) If $\mathrm{Post}_{\mu(\eta(h))}(h) \subseteq F_A(h)$ on every target prefix, then the induced outcome set is a nonempty subset of A. Conversely, if the induced outcome set is $A' \subseteq A$, the same inclusion holds at every $h \in \mathrm{Pref}(A')$.

Proof. Part (i) follows from Lemma 3.2: equality is necessary and makes the controlled and target prefix trees identical. Part (ii) is exactly the condition that one action serve every target history in each summary fiber. For (iii), local inclusion keeps every controlled prefix in $\mathrm{Pref}(A)$, whose branch set is A because A is closed. Conversely, if a retained successor at $h \in \mathrm{Pref}(A')$ lay outside $F_A(h)$, then by the definition of F_A that step admits no extension within A; totality and nonblockingness supply an infinite controlled continuation of it, which is an outcome outside A, contradicting $A' \subseteq A$. □

Only the values of η on target prefixes are constrained by exactness; its off-target values make the controlled-run semantics total. The theorem separates nonclosedness of the target, lack of a locally exact action, and incompatibility created by a summary fiber. Direct state commands introduce a fourth check.

Definition 4.3 (State locality)**.** The target A is state local when $\mathrm{last}(h) = \mathrm{last}(h')$ implies $F_A(h) = F_A(h')$ for all $h, h' \in \mathrm{Pref}(A)$. An interface is direct state based when $\mathrm{Post}_u(h)$ depends only on u and $\mathrm{last}(h)$.

Proposition 4.4 (State-locality obstruction)**.** *Suppose the interface is direct state based, $\alpha : X \to Z$, and $\eta(h) = \alpha(\mathrm{last}(h))$. If A is not state local, no selector through α realizes A exactly.*

Proof. Two target histories ending in the same state receive the same action and hence the same selected postset, which cannot equal two distinct ideal filters. □

Theorem 4.2 is a factorization statement rather than a synthesis algorithm. The target determines F_A, the interface determines which local filters are available, and the summary determines which histories share one action. For the finite state-safety target below, the ideal filter is state local and equals $E_D(x)$ at state x. Under a direct state-based interface, exact realization on a target-closed domain therefore reduces to selecting at each state an action with postset $E_D(x)$; Section 5 minimizes the number of distinct selected actions, not the size of a prescribed observation alphabet.

5. Finite state-safety targets and support covers

Assume now that X is finite. A direct restrictive interface is $\mathcal{I} = (U, (\mathrm{Post}_u)_{u\in U})$, where U is finite and nonempty and

$$\varnothing \neq \mathrm{Post}_u(x) \subseteq \mathrm{Post}(x) \qquad (x \in X, u \in U).$$

Let $D \subseteq X$ be a nonempty safety set. Its viability kernel is

$$Q_D = \nu W.\{x \in D : \mathrm{Post}(x) \cap W \neq \varnothing\}. \tag{2}$$

Assume $Q_D \neq \varnothing$, abbreviate it by Q, and define

$$E_D(x) = \mathrm{Post}(x) \cap Q \qquad (x \in Q). \tag{3}$$

Proposition 5.1 (Viability facts)**.** *Q is the greatest subset of D with a successor in itself at every state. Moreover, an infinite D-valued run exists from x exactly when $x \in Q$, and every such run remains in Q.*

Proof. This is the greatest-fixed-point characterization on the finite powerset lattice. Repeated successor choice gives the forward implication; the states on any D-valued run form a post-fixed set for the converse. □

Definition 5.2 (Implementation domain)**.** A nonempty set $K \subseteq Q$ is target closed when $E_D(x) \subseteq K$ for every $x \in K$.

For $\varnothing \neq S \subseteq Q$, let $\mathrm{Reach}_D(S)$ be the states reachable from S by finite E_D-paths.

Lemma 5.3 (Reachability characterization)**.** $\mathrm{Reach}_D(S)$ *is the least target-closed subset of Q containing S.*

Proof. Closure follows by appending one viable step; induction on path length gives minimality. □

Definition 5.4 (Safe and exact state commands)**.** A state command is a total map $c : X \to U$ and induces the postset $\mathrm{Post}_c(x) = \mathrm{Post}_{c(x)}(x)$. For $x \in X$, write $\mathrm{Runs}_P(x)$ for the infinite plant runs from x and $\mathrm{Runs}(P, c, x)$ for those satisfying $\tau_{t+1} \in \mathrm{Post}_c(\tau_t)$ at every step. The command is safe on K when

$$\mathrm{Post}_c(x) \subseteq Q \qquad (x \in K),$$

and exact on K when

$$\mathrm{Post}_c(x) = E_D(x) \qquad (x \in K).$$

Its repertoire on K is $c(K)$. Because K is nonempty, any assignment $K \to U$ extends to a total command without changing its repertoire on K.

Throughout, "safe" denotes this invariant reading, $\mathrm{Post}_c(x) \subseteq Q$ — containment in the viability kernel — rather than the weaker one-step condition $\mathrm{Post}_c(x) \subseteq D$; Proposition 5.5 justifies the convention by showing that containment in Q is equivalent to every controlled run from K remaining in D.

Proposition 5.5 (State-safety specialization)**.** *Let $K \subseteq Q$ be target closed and let $c : X \to U$ be total. The command is safe on K if and only if*

$$\mathrm{Runs}(P, c, x) \subseteq D^\omega \qquad (x \in K).$$

It is exact on K if and only if

$$\mathrm{Runs}(P, c, x) = \{\tau \in \mathrm{Runs}_P(x) : \tau_t \in D \text{ for every } t\} \qquad (x \in K).$$

Proof. If c is safe, then $\mathrm{Post}_c(x) \subseteq E_D(x) \subseteq K$ for $x \in K$, so every controlled run remains in D. If safety fails, some selected successor lies outside Q. Because the command is total and every action postset is nonempty, there is an infinite controlled continuation, and no such continuation can remain in D.

If c is exact, the controlled and viable path trees are identical. Conversely, a missing successor in $E_D(x)$ begins a viable plant run absent from $\mathrm{Runs}(P, c, x)$, while an added successor outside $E_D(x)$ begins a controlled run that cannot remain in D. □

Proposition 5.6 (Initial-domain exactness)**.** *Let $\varnothing \neq S \subseteq Q$, $K = \mathrm{Reach}_D(S)$, and $c : X \to U$ be total. Exact run-set realization by c from every state of S is equivalent to exactness of c on K.*

Proof. Exactness on K gives the required run-set equality from S. Conversely, every $x \in K$ occurs on a viable prefix from some $s \in S$. Run-set equality from s makes that prefix controlled, so x is reached under c; the local-filter argument of Lemma 3.2, applied at x, then forces every retained successor to be viable and every viable successor to be retained, that is, $\mathrm{Post}_c(x) = E_D(x)$. □

For $\star \in \{\mathsf{Safe}, \mathsf{Exact}\}$, define

$$q_\star(D, K; \mathcal{I}) = \min\{|c(K)| : c : X \to U \text{ is } \star\text{-admissible on } K\},$$

with value $+\infty$ when no such command exists. The choice of K is part of the instance. Taking $K = Q$ requires one command over the entire viability kernel; taking $K = \mathrm{Reach}_D(S)$ restricts exactness to states occurring on target runs from S. The analogous safe cover on that same K is a uniform comparator: a safety-only command from S may prune target branches before some states of K become reachable. If $K \subseteq K' \subseteq Q$ are target closed, restriction gives $q_\star(D, K; \mathcal{I}) \leq q_\star(D, K'; \mathcal{I})$.

Proposition 5.7 (Fixed observations)**.** *For a fixed observation map $o : K \to Y$ and selector $\mu : Y \to U$, the induced assignment is $\star$-admissible on K if and only if*

$$\mu(y) \in \bigcap_{x : o(x) = y} U^\star_{\mathcal{I}}(x) \qquad (y \in o(K)),$$

where $U_{\mathcal{I}}^{\mathsf{Safe}}(x) = \{u : \mathrm{Post}_u(x) \subseteq Q\}$ *and* $U_{\mathcal{I}}^{\mathsf{Exact}}(x) = \{u : \mathrm{Post}_u(x) = E_D(x)\}$. *Consequently, a* $\star$*-admissible selector through* o *exists if and only if*

$$\bigcap_{x:o(x)=y} U_{\mathcal{I}}^{\star}(x) \neq \varnothing \qquad (y \in o(K)).$$

Proof. The selector is admissible exactly when its chosen action is locally admissible at every state in each observation fiber. The existence statement follows by choosing one action from each nonempty intersection; values outside $o(K)$ may be chosen arbitrarily. □

Definition 5.8 (Action supports)**.** For $u \in U$, define

$$K_{u,\mathcal{I}}^{\mathsf{Safe}} = \{x \in K : \mathrm{Post}_u(x) \subseteq Q\}, \qquad K_{u,\mathcal{I}}^{\mathsf{Exact}} = \{x \in K : \mathrm{Post}_u(x) = E_D(x)\}.$$

A set $V \subseteq U$ with $K \subseteq \bigcup_{u\in V} K_{u,\mathcal{I}}^{\mathsf{Exact}}$ is an *equality-support cover*; its minimum size, with value $+\infty$ when none exists, is $q_{\mathsf{Exact}}(D, K; \mathcal{I})$.

A selected family covers K exactly when each state can be assigned one selected admissible action; the assignment extends to a total command without changing its repertoire on K. Hence

$$q_\star(D, K; \mathcal{I}) = \min\Big\{|V| : V \subseteq U,\ K \subseteq \bigcup_{u\in V} K_{u,\mathcal{I}}^{\star}\Big\}, \tag{C}$$

with extended value $+\infty$. Thus $q_\star$ is finite exactly when every state belongs to some $\star$-support; when finite, $1 \le q_\star \le |K|$.

Definition 5.9 (Horizon-one invariant cover)**.** A horizon-one invariant cover of K is a finite cover $\mathcal{C}$ of K by nonempty cells, together with $G : \mathcal{C} \to U$, such that $\mathrm{Post}_{G(C)}(x) \subseteq K$ for all $x \in C$. Let $r_{\mathrm{inv}}(K; \mathcal{I})$ be the minimum number of cells, with value $+\infty$ when no such cover exists.

Theorem 5.10 (Partition and invariant-cover equivalence)**.** *For every target-closed* $K \subseteq Q$*:*

(i) $q_{\mathsf{Safe}}(D, K; \mathcal{I}) = r_{\mathrm{inv}}(K; \mathcal{I})$*;*

(ii) $q_\star(D, K; \mathcal{I})$ *is the minimum number of classes in a partition of* K *such that each class admits one action that is* $\star$*-admissible at all of its states, with value* $+\infty$ *when no such partition exists.*

Proof. A support cover induces a compatible partition by assigning each state to one selected support; conversely, the actions attached to compatible partition classes form a support cover, giving part (ii). On target-closed K, $\mathrm{Post}_u(x) \subseteq Q$ is equivalent to $\mathrm{Post}_u(x) \subseteq K$. Hence a cell C labelled by u is admissible exactly when $\varnothing \neq C \subseteq K_{u,\mathcal{I}}^{\mathsf{Safe}}$. A safe-support cover therefore induces an invariant cover, while the action labels of any invariant cover form a safe-support cover. Cells carrying the same action may be merged, so the two minimum cardinalities are equal: $r_{\mathrm{inv}}(K; \mathcal{I}) = q_{\mathsf{Safe}}(D, K; \mathcal{I})$, proving part (i). □

Library enlargement cannot increase either cover number, and $q_\star = 1$ exactly when one action has support K.

Definition 5.11 (Statewise patching closure)**.** The interface is statewise patching closed on K when, for every map $f : K \to U$, some action $u_f \in U$ satisfies $\mathrm{Post}_{u_f}(x) = \mathrm{Post}_{f(x)}(x)$ for every $x \in K$.

Proposition 5.12 (Patching collapse)**.** *If* $\mathcal{I}$ *is statewise patching closed on* K*, then finite* $q_\star(D, K; \mathcal{I})$ *implies* $q_\star(D, K; \mathcal{I}) = 1$.

Proof. Choose one locally admissible action at each state and patch those local postsets into one global library action. □

Patching closure isolates why a nontrivial repertoire can arise: the declared library cannot package independently chosen local postsets into one global action. Failure of patching closure is necessary, though not sufficient, for a finite cover number to exceed one. To see that it is not sufficient, let

$$X = D = K = \{x_1, x_2\}, \qquad \mathrm{Post}(x_1) = \mathrm{Post}(x_2) = K.$$

Then $Q = K$ and $E_D(x_i) = K$ for $i = 1, 2$, so K is target closed. Take the library $\{u_0, u_1, u_2\}$ with

	x_1	x_2
Post_{u_0}	K	K
Post_{u_1}	$\{x_1\}$	K
Post_{u_2}	K	$\{x_2\}$

(where each entry is the selected postset at the column state). The action u_0 is exact at both states, so $q_{\mathsf{Exact}} = q_{\mathsf{Safe}} = 1$. However, the assignment $x_1 \mapsto u_1$, $x_2 \mapsto u_2$ has postset pair $(\{x_1\}, \{x_2\})$, which no action in the library realizes. Hence the interface is not statewise patching closed.

5.1. **Pointwise nested interfaces.**

Definition 5.13 (Pointwise nested interface)**.** The interface is pointwise nested on K when $U = \{u_1, \ldots, u_m\}$ can be indexed so that

$$\mathrm{Post}_{u_1}(x) \subseteq \mathrm{Post}_{u_2}(x) \subseteq \cdots \subseteq \mathrm{Post}_{u_m}(x) \qquad (x \in K).$$

Theorem 5.14 (Exact-action intervals under a nested interface)**.** *Suppose $\mathcal{I}$ is pointwise nested on K.*

(i) If $q_{\mathsf{Safe}}(D, K; \mathcal{I}) < +\infty$, then $q_{\mathsf{Safe}}(D, K; \mathcal{I}) = 1$.

(ii) For $x \in K$, let

$$I_x = \{j \in \{1, \ldots, m\} : \mathrm{Post}_{u_j}(x) = E_D(x)\}.$$

If $I_x \neq \varnothing$, then it is an integer interval $[\ell_x, r_x] \cap \{1, \ldots, m\}$, and

$$\begin{aligned} &\mathrm{Post}_{u_j}(x) \subsetneq E_D(x) && (j < \ell_x), \\ &\mathrm{Post}_{u_j}(x) = E_D(x) && (\ell_x \le j \le r_x), \\ &\mathrm{Post}_{u_j}(x) \supsetneq E_D(x) \quad \textit{and} \quad \mathrm{Post}_{u_j}(x) \not\subseteq Q && (j > r_x). \end{aligned} \tag{N}$$

(iii) If $q_{\mathsf{Exact}} < +\infty$, then q_{Exact} is the minimum number of indices meeting every interval I_x. It equals the maximum number of pairwise disjoint intervals and is found by repeatedly choosing the smallest right endpoint among the remaining intervals.

Proof. For (i), local safety at x for some u_{j_x} implies safety of u_1 there by nestedness. For (ii), any index between two exact indices is exact. Indices below the first exact one are strict safe under-approximations; an index above the last exact one adds a successor outside Q because $E_D(x) = \mathrm{Post}(x) \cap Q$. Parts (i)–(ii) give the interface-specific structure. Part (iii) applies Gallai's classical interval-transversal theorem, which Gallai did not publish and which was first printed by Hajnal and Surányi: the smallest-right-endpoint greedy points form a transversal, while the witness intervals chosen at successive iterations are pairwise disjoint (Hajnal and Surányi, 1958). □

Once the intervals are formed, the right-endpoint algorithm runs in $O(|K|\log|K|)$ time.

5.2. **Global event-disable masks and the event-effect boundary.** Let $\Gamma = \Gamma_c \dot\cup \Gamma_{uc}$ be a finite event alphabet.

Definition 5.15 (Global event-disable mask interface)**.** Suppose the plant has a total deterministic event realization $\delta_P : X \times \Gamma \to X$ with

$$\mathrm{Post}(x) = \{\delta_P(x,\gamma) : \gamma \in \Gamma\}.$$

Thus every event is available at every state; unavailable-event semantics are outside this model. A mask is a set $B \subseteq \Gamma_c$ with $\Gamma \setminus B \neq \varnothing$. It disables the controllable events in B and retains every event in $\Gamma \setminus B$, including all uncontrollable events. Its postset is

$$\mathrm{Post}_{u_B}(x) = \{\delta_P(x,\gamma) : \gamma \in \Gamma \setminus B\}.$$

A nonempty finite library

$$\varnothing \neq \mathcal{B} \subseteq \{B \subseteq \Gamma_c : \Gamma \setminus B \neq \varnothing\}$$

induces the interface $\mathcal{I}_{\mathcal{B}}$.

For $x \in K$, define the controllable hazard set and event-effect fibers

$$H_D(x) = \{\gamma \in \Gamma_c : \delta_P(x,\gamma) \notin Q\}, \qquad \Phi_x(y) = \{\gamma \in \Gamma : \delta_P(x,\gamma) = y\}.$$

Distinct events in one fiber are indistinguishable to the present exactness objective, which compares state-successor sets rather than event-labelled traces.

Theorem 5.16 (Global-mask effect-fiber characterization)**.** *Let $K \subseteq Q$ be target closed and let $\mathcal{I}_{\mathcal{B}}$ be a global mask interface. Suppose*

$$\delta_P(x,\gamma) \in Q \qquad (x \in K, \gamma \in \Gamma_{uc}). \tag{UQ}$$

Then, for every $x \in K$ and $B \in \mathcal{B}$,

$$u_B \text{ is safe at } x \iff H_D(x) \subseteq B, \tag{M$\subseteq$}$$

$$u_B \text{ is exact at } x \iff H_D(x) \subseteq B \text{ and } \Phi_x(y) \setminus B \neq \varnothing \quad (\forall y \in E_D(x)). \tag{MF=}$$

Proof. Under (UQ), only controllable events can leave Q, so safety is equivalent to disabling every hazardous controllable event. Under that condition the selected postset is contained in $E_D(x)$. The reverse inclusion holds exactly when each viable successor is still produced by at least one enabled event, which is the fiber condition. □

If (UQ) fails at any implementation state, every mask retains a transition outside Q, so both cover numbers are $+\infty$. Equation (M$\subseteq$) also gives, without any injectivity assumption,

$$K^{\mathsf{Safe}}_{u_B,\mathcal{I}_{\mathcal{B}}} = \{x \in K : H_D(x) \subseteq B\}, \qquad q_{\mathsf{Safe}} = 1 \iff \bigcup_{x\in K} H_D(x) \subseteq B \text{ for some } B \in \mathcal{B}. \tag{MS}$$

When $\Gamma_{uc} \neq \varnothing$, the all-controllable-disable mask $B = \Gamma_c$ is safe throughout K whenever it belongs to the library.

Example 5.17 (Reading H_D and Φ_x from a tool-gating description)**.** Consider an agent whose controllable tool events are $\Gamma_c = \{\mathsf{search}, \mathsf{send}, \mathsf{purge}\}$ and whose single uncontrollable environment response is $\Gamma_{uc} = \{\mathsf{reply}\}$. States record task status, $X = \{\mathsf{draft}, \mathsf{sent}, \bot\}$, with

safety set $D = \{\mathsf{draft}, \mathsf{sent}\}$ and $\perp$ an irrecoverable state (for instance, required context has been destroyed). The event realization is

$$\text{at } \mathsf{draft}: \quad \mathsf{search} \mapsto \mathsf{draft}, \quad \mathsf{send} \mapsto \mathsf{sent}, \quad \mathsf{purge} \mapsto \perp, \quad \mathsf{reply} \mapsto \mathsf{draft},$$
$$\text{at } \mathsf{sent}: \quad \mathsf{search} \mapsto \mathsf{sent}, \quad \mathsf{send} \mapsto \mathsf{sent}, \quad \mathsf{purge} \mapsto \perp, \quad \mathsf{reply} \mapsto \mathsf{draft},$$

with every event self-looping at $\perp$. The uncontrollable reply keeps both task states viable, so $Q = \{\mathsf{draft}, \mathsf{sent}\}$ and $E_D(x) = \{\mathsf{draft}, \mathsf{sent}\}$ at each $x \in Q$. The hazard sets and nonempty effect fibers are read directly off the table:

$$H_D(\mathsf{draft}) = H_D(\mathsf{sent}) = \{\mathsf{purge}\},$$
$$\Phi_{\mathsf{draft}}(\mathsf{draft}) = \{\mathsf{search}, \mathsf{reply}\}, \qquad \Phi_{\mathsf{draft}}(\mathsf{sent}) = \{\mathsf{send}\},$$
$$\Phi_{\mathsf{sent}}(\mathsf{sent}) = \{\mathsf{search}, \mathsf{send}\}, \qquad \Phi_{\mathsf{sent}}(\mathsf{draft}) = \{\mathsf{reply}\},$$

since purge is the only controllable event leaving Q. A permission profile is a mask $B \subseteq \Gamma_c$; the single profile $B = \{\mathsf{purge}\}$ satisfies $H_D(x) \subseteq B$ and leaves every viable fiber inhabited, so by (MF=) it is exact at both states and $q_{\mathsf{Exact}} = 1$: one profile that disables the lone hazardous tool preserves the intended behaviour exactly. The duplicated effect — search and reply both yield draft — is the noninjective phenomenon of Theorem 5.19; here it makes search redundant for exactness at draft, since reply already realizes that successor.

Corollary 5.18 (Successor-injective effects: the exact hazard-set formula)**.** *Under the hypotheses of Theorem 5.16, suppose in addition that $\gamma \mapsto \delta_P(x, \gamma)$ is injective on Γ for every $x \in K$. Then*

$$u_B \text{ is exact at } x \iff H_D(x) = B. \tag{M=}$$

Consequently, with $\mathcal{H}_K = \{H_D(x) : x \in K\}$,

$$q_{\mathsf{Exact}}(D, K; \mathcal{I}_{\mathcal{B}}) = \begin{cases} |\mathcal{H}_K|, & \mathcal{H}_K \subseteq \mathcal{B}, \\ +\infty, & \text{otherwise.} \end{cases} \tag{MR}$$

Proof. Theorem 5.16 gives $H_D(x) \subseteq B$. If $\gamma \in B \setminus H_D(x)$, its successor lies in $E_D(x)$ and injectivity makes its fiber the singleton $\{\gamma\}$, contradicting the fiber condition. Hence exactness implies $B = H_D(x)$.

Conversely, suppose $B = H_D(x)$. For every $y \in E_D(x)$, choose $\gamma \in \Phi_x(y)$, which is possible because $y \in \mathrm{Post}(x)$. Since $\delta_P(x, \gamma) = y \in Q$, the event γ is either uncontrollable or a nonhazardous controllable event. Thus $\gamma \notin H_D(x) = B$, so every viable fiber retains an enabled event. Theorem 5.16 therefore makes u_B exact at x.

Moreover, $H_D(x)$ is a mask in the sense of Definition 5.15 for every $x \in K$: the nonempty set $E_D(x)$ has an event generator outside $H_D(x)$. Exact feasibility therefore reduces to membership of each represented hazard set in the declared library, and exact supports are the disjoint hazard-set classes. □

Theorem 5.19 (Arbitrary supports beyond successor injectivity)**.** *Let K be finite and nonempty and let $S_1, \ldots, S_m \subseteq K$, $m \geq 1$. There is a total deterministic plant with state set $X = K \cup \{\perp\}$, target $D = K$, viability kernel $Q_D = K$, one uncontrollable event, and m global masks such that*

$$K^{\mathsf{Safe}}_{u_{B_j}, \mathcal{I}_{\mathcal{B}}} = K^{\mathsf{Exact}}_{u_{B_j}, \mathcal{I}_{\mathcal{B}}} = S_j \qquad (1 \leq j \leq m).$$

Every controllable event at $x \in K$ can be chosen to have successor either x or $\perp$, and every mask retains only the uncontrollable event and one controllable event. If $\bigcup_j S_j = K$, successor injectivity fails at every state of K.

Proof. Take $\Gamma_{uc} = \{\tau\}$ and $\Gamma_c = \{e_1, \dots, e_m\}$. Let every event self-loop at $\perp$, and for $x \in K$ set

$$\delta_P(x, \tau) = x, \qquad \delta_P(x, e_j) = \begin{cases} x, & x \in S_j, \\ \perp, & x \notin S_j. \end{cases}$$

The uncontrollable self-loop gives $Q_D = K$ and $E_D(x) = \{x\}$. Use $B_j = \Gamma_c \setminus \{e_j\}$. The retained postset is $\{x\}$ on S_j and $\{x, \perp\}$ outside S_j, so both supports are exactly S_j. If the family covers K, some retained controllable event duplicates the uncontrollable self-loop at every implementation state. □

Event-effect regime	Local exact-mask condition	Exact-support consequence
Successor-injective	$B = H_D(x)$	Exact supports are disjoint hazard classes; the feasible optimum is the number of distinct represented hazard sets.
Duplicate effects allowed	$H_D(x) \subseteq B$ and every viable effect fiber retains an event	Arbitrary finite exact-support systems are realizable; exact repertoire minimization is NP-hard.

TABLE 1. Exact-support boundary for total deterministic global masks with viable uncontrollable transitions.

Global disable patterns retaining every uncontrollable event are standard in supervisory control (Ramadge and Wonham, 1987; Wonham and Ramadge, 1987). Table 1 summarizes the structural distinction: successor injectivity rigidifies exact supports, while duplicate effects permit arbitrary exact-support geometry. (In the tool-gating reading, duplicate effects arise when distinct tools can produce the same state — e.g. two retrieval tools returning the same result — which is the regime where exact repertoire minimization becomes hard.) Safety does not collapse under successor injectivity: its supports remain the containment sets in (MS), and the next section shows that safety remains a general covering problem even on the successor-injective side.

5.3. **Optimization and the Set Cover boundary.** Throughout this subsection, the finite plant, action postsets, and mask library are given explicitly. Once the supports in Definition 5.8 are formed, equation (C) is a Set Cover instance. Theorem 5.19 embeds arbitrary support systems into noninjective masks. The following construction gives the complementary safety result under successor-injective effects.

Proposition 5.20 (Successor-injective realization of safe Set Cover)**.** *Let V be finite and nonempty and let $F_1, \dots, F_m \subseteq V$ cover V. There is a total deterministic global-mask instance with one uncontrollable event, successor-injective event effects at every implementation state, and masks $B_1, \dots, B_m$ such that, for a padding set W contained in every safe support,*

$$K^{\mathsf{Safe}}_{u_{B_j}, \mathcal{I}_{\mathcal{B}}} = F_j \cup W \qquad (1 \le j \le m).$$

Hence the minimum safe repertoire equals the Set Cover optimum of $(V, \{F_j\}_{j=1}^m)$.

Proof. Let $\Gamma_{uc} = \{\tau\}$, $\Gamma_c = \{g_1, \dots, g_m\}$, $W = \{p_0, p_1, \dots, p_m\}$, and $O = \{o_1, \dots, o_m\}$, with all sets disjoint. Put $X = V \cup W \cup O$, $D = K = V \cup W$, and define, for $x \in V$,

$$\delta_P(x, \tau) = p_0, \qquad \delta_P(x, g_j) = \begin{cases} p_j, & x \in F_j, \\ o_j, & x \notin F_j. \end{cases}$$

For $p_i \in W$, set $\delta_P(p_i, \tau) = p_0$ and $\delta_P(p_i, g_j) = p_j$; complete the transitions on O arbitrarily. At every state of K, the $m + 1$ events have distinct successors. Since τ reaches p_0 from V and all transitions from W remain in W, $Q_D = K$ and (UQ) holds. The hazards are

$$H_D(x) = \{g_j : x \notin F_j\} \quad (x \in V), \qquad H_D(p_i) = \varnothing.$$

Use $B_j = \Gamma_c \setminus \{g_j\}$. By (M$\subseteq$), B_j is safe at $x \in V$ exactly when $g_j \notin H_D(x)$, equivalently $x \in F_j$, and it is safe at every padding state. Thus its support is $F_j \cup W$. Because every chosen mask covers all of W, covering K is equivalent, with identical cardinality, to covering V by the F_j. □

Corollary 5.21 (Injective safe-mask hardness)**.** *The threshold problem $q_{\mathsf{Safe}}(D, K; \mathcal{I}_{\mathcal{B}}) \le k$ is NP-complete even for total deterministic global masks satisfying (UQ), successor injectivity on K, and the restriction that every mask retains one uncontrollable and exactly one controllable event.*

Proof. Membership in NP follows by checking at most k masks against all states. Proposition 5.20 is an optimum-preserving polynomial reduction from Set Cover (Karp, 1972). □

Example 5.22 (Successor injectivity does not simplify safety)**.** Let $\Gamma_{uc} = \varnothing$, so (UQ) holds automatically, and let $\Gamma_c = \{a, b, c\}$ and $K = \{x_1, x_2, x_3\}$, with hazards $\{a\}$, $\{b\}$, and $\{c\}$, respectively. Take the two-mask library $B_1 = \{a, b\}$ and $B_2 = \{b, c\}$. The support calculation is

state	$H_D(x)$	safe masks	exact masks under successor injectivity
x_1	$\{a\}$	B_1	$\varnothing$
x_2	$\{b\}$	B_1, B_2	$\varnothing$
x_3	$\{c\}$	B_2	$\varnothing$

Thus $K^{\mathsf{Safe}}_{u_{B_1}} = \{x_1, x_2\}$ and $K^{\mathsf{Safe}}_{u_{B_2}} = \{x_2, x_3\}$, so $q_{\mathsf{Safe}} = 2$. Corollary 5.18 requires $B = H_D(x)$ for exactness, and neither library mask equals a local hazard set; hence $q_{\mathsf{Exact}} = +\infty$.

One realization takes $X = K \cup \{\bot\}$, $D = K$, and

$$(\delta_P(x_1, a), \delta_P(x_1, b), \delta_P(x_1, c)) = (\bot, x_1, x_2),$$
$$(\delta_P(x_2, a), \delta_P(x_2, b), \delta_P(x_2, c)) = (x_2, \bot, x_3),$$
$$(\delta_P(x_3, a), \delta_P(x_3, b), \delta_P(x_3, c)) = (x_3, x_1, \bot),$$

with every event self-looping at $\bot$. The stated hazards follow, and the effects are successor-injective on K. Such effects simplify exact but not safe minimization: exact supports are disjoint hazard classes, whereas the safe supports overlap as above.

For either objective, an exact optimum is given by the standard binary Set Cover formulation

$$\min \sum_{u \in U} z_u \quad \text{subject to} \quad \sum_{u : x \in K^{\star}_{u,\mathcal{I}}} z_u \ge 1 \ (x \in K), \qquad z_u \in \{0, 1\}.$$

Infeasibility corresponds to $+\infty$.

Corollary 5.23 (Support-cover complexity across the event-effect boundary)**.** *Under this explicit-table encoding, both threshold problems $q_{\mathsf{Safe}} \le k$ and $q_{\mathsf{Exact}} \le k$ are NP-complete in general. More specifically:*

(i) under successor-injective global masks satisfying (UQ), the exact optimum is computable in polynomial time by Corollary 5.18, while the safe threshold problem remains NP-complete by Corollary 5.21;

(ii) without successor injectivity, both threshold problems are NP-complete even when $D = K = Q_D$, $E_D(x) = \{x\}$, one uncontrollable event self-loops on K, each controllable event has successor in $\{x, \perp\}$, and each mask retains that uncontrollable event and one controllable event;

(iii) on feasible instances, greedy support cover is an $H_{|K|} \le 1 + \ln |K|$ approximation. In the noninjective family of Theorem 5.19, for every fixed $0 < \varepsilon < 1$, neither objective admits a polynomial-time $(1 - \varepsilon) \ln |K|$ approximation unless P = NP*.*

Proof. General NP membership is immediate from explicit support checking. Part (i) combines Corollaries 5.18 and 5.21. For part (ii), apply Theorem 5.19 to a Set Cover instance $(V, \{F_j\})$ with $K = V$ and $S_j = F_j$; safe and exact supports both equal F_j, preserving the optimum. Part (iii) transfers the classical greedy bound (Chvátal, 1979) and the $(1 - \varepsilon) \ln n$ hardness threshold (Moshkovitz, 2015; Dinur and Steurer, 2014), with $n = |K|$. □

Procedure 5.24 (Finite fixed-interface support-cover reduction and solution)**.** The input consists of explicit plant and action tables, an objective $\star \in \{\mathsf{Safe}, \mathsf{Exact}\}$, and either an initial set S or a declared domain K. The output is an optimum value and witness when an exact Set Cover or integer-programming solver is used, or a greedy witness and bound otherwise. When S and the safety objective are supplied, the output is the uniform comparator on $\mathrm{Reach}_D(S)$, not the minimum repertoire for safety from S alone.

(1) Compute the viability kernel Q_D by repeatedly deleting $x \in Q$ with $\mathrm{Post}(x) \cap Q = \varnothing$, starting from $Q \leftarrow D$.
(2) Set $E_D(x) = \mathrm{Post}(x) \cap Q$ for $x \in Q$.
(3) If S is supplied, reject the instance when $S = \varnothing$. If $S \not\subseteq Q$, return $+\infty$; otherwise set $K = \mathrm{Reach}_D(S)$. If K is supplied, reject the instance unless it is a nonempty target-closed subset of Q.
(4) Form the requested supports by testing $\mathrm{Post}_u(x) \subseteq Q$ and $\mathrm{Post}_u(x) = E_D(x)$ for every $x \in K$ and $u \in U$. If some state has no locally admissible action, return $+\infty$.
(5) For a pointwise nested interface, return $(1, \{u_1\})$ for safety. For exactness, form the intervals I_x and apply the right-endpoint algorithm.
(6) For a global mask interface, first test (UQ); if it fails, return $+\infty$. For exactness with successor-injective effects, return $+\infty$ unless $\mathcal{H}_K \subseteq \mathcal{B}$, and otherwise select the masks in $\mathcal{H}_K$. Without successor injectivity, form exact supports using (MF=). For safety, form the containment supports in (MS), regardless of successor injectivity.
(7) Pass the remaining support cover to an exact Set Cover or integer-programming solver using the binary formulation above, or apply greedy Set Cover for the stated approximation guarantee.

The viability, support-formation, nested, and successor-injective exact-mask branches are polynomial in the explicit table size; every remaining branch exposes a Set Cover instance.

Under successor-injective effects, represented hazard sets determine exact feasibility, while duplicate effects restore arbitrary exact-support geometry.

5.4. **Tight separations under restricted interfaces.**

Theorem 5.25 (Restricted-interface safety–exactness gap)**.** *If* $q_{\mathsf{Exact}}(D, K; \mathcal{I}) < +\infty$*, then*

$$1 \le q_{\mathsf{Safe}}(D, K; \mathcal{I}) \le q_{\mathsf{Exact}}(D, K; \mathcal{I}) \le |K|.$$

For every $n \ge 2$*, the upper separation*

$$q_{\mathsf{Safe}}(D, K; \mathcal{I}) = 1, \qquad q_{\mathsf{Exact}}(D, K; \mathcal{I}) = n, \qquad |K| = n,$$

with every nonempty exact support of cardinality one, occurs in both:

(a) a pointwise nested interface generated by prefixes of deterministic successor channels;
(b) a successor-injective global-mask interface retaining every uncontrollable event, with one all-controllable-disable mask and n *masks that each disable one controllable event.*

Proof. Every exact support is contained in the corresponding safe support. Local feasibility and the choice of at most one witness action per state give the remaining universal bounds.

For (a), let $K = D = \{x_1, \dots, x_n\}$ and $X = K \cup \{\bot\}$. Define channels

$$a_j(x_i) = \begin{cases} x_j, & j \le i, \\ \bot, & j > i, \end{cases} \qquad a_j(\bot) = \bot.$$

Set $\mathrm{Post}(x) = \{a_j(x) : 1 \le j \le n\}$ and $\mathrm{Post}_{u_t}(x) = \{a_j(x) : 1 \le j \le t\}$. Then $Q_D = K$ and $E_D(x_i) = \{x_1, \dots, x_i\}$. At x_i, u_t is a strict safe under-approximation for $t < i$, exact for $t = i$, and unsafe for $t > i$. Thus $K^{\mathsf{Exact}}_{u_i, \mathcal{I}} = \{x_i\}$, while u_1 is safe on all of K.

For (b), again take $K = D = \{x_1, \dots, x_n\}$, $X = K \cup \{\bot\}$, $\Gamma_{uc} = \{\tau\}$, and $\Gamma_c = \{a_1, \dots, a_n\}$. Set

$$\delta_P(x_i, \tau) = x_i, \qquad \delta_P(x_i, a_i) = \bot, \qquad \delta_P(x_i, a_j) = x_j \ (j \ne i),$$

and self-loop every event at $\bot$. Effects are successor-injective on K, $Q_D = K$, $E_D(x_i) = K$, and $H_D(x_i) = \{a_i\}$. Use the library $\{\Gamma_c, \{a_1\}, \dots, \{a_n\}\}$. The all-controllable-disable mask Γ_c is safe everywhere, while Corollary 5.18 makes $\{a_i\}$ exact only at x_i. □

The two constructions show that polynomial-time optimization does not imply a small exact repertoire.

5.5. **Why invariant, equality, and control covers are not interchangeable.** The three cover notions use similar language but optimize different objects. A horizon-one invariant-cover cell needs one action whose image remains in K. An equality-support cell requires literal equality with E_D. A supervisory control cover groups supervisor states that are control compatible and event-successor consistent, and its cardinality is the number of states in an induced reduced supervisor (Vaz and Wonham, 1986; Su and Wonham, 2004, 2018).

For the next proposition, use the Su–Wonham convention. At supervisor state z, let $E(z)$ be enabled events, $D(z)$ plant-feasible disabled events, $M(z)$ the supervisor marking indicator, and let $T(z) = 1$ exactly when some string that reaches z in the supervisor also reaches a marked state in the plant. Two states are control compatible when

$$E(z) \cap D(z') = E(z') \cap D(z) = \varnothing, \qquad T(z) = T(z') \Rightarrow M(z) = M(z').$$

A control cover may overlap, but every cell contains pairwise compatible states and, for each event, all defined successor states from that cell must lie in one successor cell. Let $r_{\mathrm{cc}}(G,S)$ be its minimum size.

Proposition 5.26 (Cover objectives retain different information)**.** *(i) For every $n \geq 2$, a fixed-interface instance has*

$$r_{\mathrm{inv}}(K;\mathcal{I}) = q_{\mathsf{Safe}}(D,K;\mathcal{I}) = 1, \qquad q_{\mathsf{Exact}}(D,K;\mathcal{I}) = n.$$

The instance may be chosen from either family in Theorem 5.25.

(ii) There is a deterministic one-event plant–supervisor pair (G,S) with $r_{\mathrm{cc}}(G,S) = 3$ such that, after erasing event labels and marking, its unlabelled transition graph viewed as a one-action equality-support instance has $q_{\mathsf{Exact}} = 1$.

Proof. Part (i) combines Theorems 5.10 and 5.25.

For (ii), let $z_0 \xrightarrow{a} z_1 \xrightarrow{a} z_2 \xrightarrow{a} z_0$ be both the plant and supervisor transition cycle, with a observable and uncontrollable. Mark every plant state but only z_0, z_1 in the supervisor. Because every plant state is marked, $T(z_i) = 1$ for $i = 0,1,2$, whereas $M(z_0) = M(z_1) = 1$ and $M(z_2) = 0$. There is no enabling–disabling conflict, but the marking condition prevents z_2 from sharing a cell with either marked state. Although z_0 and z_1 are control compatible, they cannot share a control-cover cell: their a-successors are z_1 and z_2, which would have to belong to one successor cell despite being incompatible. Hence every cell is a singleton and $r_{\mathrm{cc}} = 3$.

After erasing the event label and marking, take $D = K = \{z_0, z_1, z_2\}$ and one action u with $\mathrm{Post}_u(z_i) = \{z_{i+1 \bmod 3}\} = E_D(z_i)$. Its exact support is all of K, so $q_{\mathsf{Exact}} = 1$. □

Part (i) shows that replacing equality by invariance changes the local predicate and optimization problem. Part (ii) shows that the equality-support cardinality of the unlabelled successor system does not determine the control-cover cardinality of the original marked, event-labelled supervisor. Control covers retain compatibility and successor-cell information that the equality-support instance intentionally discards.

6. Related work and scope

Safety as topological closedness and prefix-detectable violation is classical (Alpern and Schneider, 1985, 1987). Truncation monitors, edit automata, suppression monitors, shields, and modern safety filters grant different intervention powers and generally do not induce the same outcome sets (Schneider, 2000; Ligatti et al., 2005; Bloem et al., 2015; Hsu et al., 2024). The arbitrary successor-subset mediator in Sections 2–4 is only the semantic baseline; the optimization begins after a fixed action library is declared. Runtime guardrails for AI agents that select among predeclared tool-permission profiles are a natural instance of this fixed-interface setting, and inherit the support-cover classification unchanged.

State-partition-based supervisory control restricts a supervisor to the original automaton states or reachable Petri-net markings and separates legal from illegal states, enabling legal-to-legal transitions and disabling legal-to-illegal transitions (Butez et al., 2014). That work synthesizes safe, nonblocking state-based supervisors. Here E_D is fixed first, each library element is a predeclared global postset map, and the objective counts how many such maps are selected on a declared plant-state domain.

Ramadge–Wonham supervisory control supplies the standard semantics of control patterns that retain uncontrollable events (Ramadge and Wonham, 1987; Wonham and Ramadge, 1987). Supervisor reduction minimizes a behavior-preserving quotient subject to control compatibility and transition consistency (Vaz and Wonham, 1986; Su and Wonham, 2004, 2018). Equality-support optimization constructs no quotient automaton and does not count supervisor states. Bisimilarity-enforcing supervisory control is closer to branching preservation, but it synthesizes supervisors under a behavioral-equivalence objective rather than compressing a fixed identity-state successor relation through a predeclared library (Takai, 2019).

Permissive strategies and control envelopes represent sets of winning or safe choices (Bernet et al., 2002; Anand et al., 2023; Kabra et al., 2024, 2026). Those frameworks synthesize a family of admissible choices relative to a game or control objective. Here the target relation E_D is fixed before compression, every action is a predeclared global postset map, and the objective counts selected maps. The comparison is between synthesis of a permissive policy representation and compression of an already fixed exact successor relation.

Horizon-one safe supports coincide with a finite invariant-cover comparator, whereas dynamic information-rate notions such as invariance entropy measure different resources (Colonius and Kawan, 2009; Gao et al., 2021). The equality-support objective changes the local predicate from inclusion to literal equality, which is why Theorem 5.25 separates the two cardinalities on the same plant, domain, and library. A control cover additionally preserves event-labelled successor-cell consistency and marking compatibility; Proposition 5.26 shows that these data are not recoverable from equality-support cardinality alone.

For pointwise nested interfaces, the derivation of exact-action intervals is model specific; the interval transversal min–max theorem and greedy algorithm are classical (Hajnal and Surányi, 1958). The common action order is strong, but it isolates a useful phenomenon: a globally ordered library may still require one exact action per state while safety collapses to the least action.

The exactness notion is equality of *state-run* branch sets and intentionally quotients events with the same successor. Theorem 5.19 would not by itself preserve event-labelled traces, event multiplicity, or stronger event-sensitive branching semantics. The finite theory assumes a target-closed domain, explicit tables, and memoryless selection from a fixed library; the mask results additionally assume viable uncontrollable transitions and the totality of Definition 5.15, under which every event is available at every state and is removed only by an explicit mask.

Totality is a modelling choice matched to the intended regime rather than a technical artifact: in both the firmware-interlock and tool-gating instances the interface exposes a full command or tool surface, and unavailability at a state is expressed *through* the selected disable pattern rather than by the event being physically absent. Partial event availability, meaning that some commands are absent at some states independently of any mask, lies outside the present model, although the pointwise effect-fiber equivalence extends directly by restricting hazards and fibers to the events available at the state, requiring every available uncontrollable event to be viable, and retaining at least one available, non-disabled event in every viable successor fiber. What is not developed here is the support geometry and complexity induced by state-dependent availability; guards, costs, compositional constraints, and observation restrictions may likewise produce intermediate support classes.

Finally, q_{Safe} and q_{Exact} are library-relative selected-action counts, not intrinsic sensor, memory, communication, or information lower bounds. Successor-injective effects make exact-mask minimization tractable, duplicate effects encode arbitrary Set Cover instances, and safe minimization remains NP-hard even under successor injectivity.

7. Conclusion

Exact branch-set preservation through a fixed interface becomes a local equality requirement between selected and viable successor sets. The interface-induced support geometry is decisive: pointwise nesting yields exact-action intervals; successor-injective mask effects collapse exact supports to hazard classes while safe supports remain overlapping and NP-hard to minimize; duplicate effects recover arbitrary support systems. Both restricted families attain the maximal separation among instances with finite q_{Exact}, namely $q_{\mathsf{Safe}} = 1$ and $q_{\mathsf{Exact}} = |K|$.

References

B. Alpern and F. B. Schneider. Defining liveness. *Information Processing Letters*, 21(4):181–185, 1985. doi:10.1016/0020-0190(85)90056-0.

B. Alpern and F. B. Schneider. Recognizing safety and liveness. *Distributed Computing*, 2(3):117–126, 1987. doi:10.1007/BF01782772.

A. Anand, S. P. Nayak, and A.-K. Schmuck. Synthesizing permissive winning strategy templates for parity games. In *CAV 2023*, LNCS 13964, pp. 436–458, 2023. doi:10.1007/978-3-031-37706-8_22.

J. Bernet, D. Janin, and I. Walukiewicz. Permissive strategies: From parity games to safety games. *RAIRO–Theoretical Informatics and Applications*, 36(3):261–275, 2002. doi:10.1051/ita:2002013.

R. Bloem, B. Könighofer, R. Könighofer, and C. Wang. Shield synthesis: Runtime enforcement for reactive systems. In *TACAS 2015*, LNCS 9035, pp. 533–548, 2015. doi:10.1007/978-3-662-46681-0_51.

A. Butez, S. Lafortune, and Y. Wang. State-partition-based control of discrete event systems for enforcement of regular language specifications. *IFAC Proceedings Volumes*, 47(3):2414–2421, 2014. doi:10.3182/20140824-6-ZA-1003.00646.

V. Chvátal. A greedy heuristic for the set-covering problem. *Mathematics of Operations Research*, 4(3):233–235, 1979. doi:10.1287/moor.4.3.233.

F. Colonius and C. Kawan. Invariance entropy for control systems. *SIAM Journal on Control and Optimization*, 48(3):1701–1721, 2009. doi:10.1137/080713902.

I. Dinur and D. Steurer. Analytical approach to parallel repetition. In *STOC 2014*, pp. 624–633, 2014. doi:10.1145/2591796.2591884.

Y. Gao, M. Cannon, L. Xie, and K. H. Johansson. Invariant cover: Existence, cardinality bounds, and computation. *Automatica*, 129:109588, 2021. doi:10.1016/j.automatica.2021.109588.

A. Hajnal and J. Surányi. Über die Auflösung von Graphen in vollständige Teilgraphen. *Annales Universitatis Scientiarum Budapestinensis de Rolando Eötvös Nominatae, Sectio Mathematica*, 1:113–121, 1958.

K.-C. Hsu, H. Hu, and J. F. Fisac. The safety filter: A unified view of safety-critical control in autonomous systems. *Annual Review of Control, Robotics, and Autonomous Systems*, 7:47–72, 2024. doi:10.1146/annurev-control-071723-102940.

A. Kabra, J. Laurent, S. Mitsch, and A. Platzer. CESAR: Control envelope synthesis via angelic refinements. In *TACAS 2024*, LNCS 14570, pp. 144–164, 2024. doi:10.1007/978-3-031-57246-3_9.

A. Kabra, J. Laurent, S. Mitsch, and A. Platzer. Hybrid game control envelope synthesis. *Proceedings of the ACM on Programming Languages*, 10(OOPSLA1):850–876, 2026. doi:10.1145/3798230.

R. M. Karp. Reducibility among combinatorial problems. In R. E. Miller and J. W. Thatcher, eds., *Complexity of Computer Computations*, pp. 85–103. Plenum Press, 1972. doi:10.1007/978-1-4684-2001-2_9.

J. Ligatti, L. Bauer, and D. Walker. Edit automata: Enforcement mechanisms for run-time security policies. *International Journal of Information Security*, 4(1–2):2–16, 2005. doi:10.1007/s10207-004-0046-8.

D. Moshkovitz. The Projection Games Conjecture and the NP-hardness of $\ln n$-approximating Set Cover. *Theory of Computing*, 11(7):221–235, 2015. doi:10.4086/toc.2015.v011a007.

P. J. Ramadge and W. M. Wonham. Supervisory control of a class of discrete event processes. *SIAM Journal on Control and Optimization*, 25(1):206–230, 1987. doi:10.1137/0325013.

F. B. Schneider. Enforceable security policies. *ACM Transactions on Information and System Security*, 3(1):30–50, 2000. doi:10.1145/353323.353382.

R. Su and W. M. Wonham. Supervisor reduction for discrete-event systems. *Discrete Event Dynamic Systems*, 14(1):31–53, 2004. doi:10.1023/B:DISC.0000005009.40749.b6.

R. Su and W. M. Wonham. What information really matters in supervisor reduction? *Automatica*, 95:368–377, 2018. doi:10.1016/j.automatica.2018.06.004.

S. Takai. Bisimilarity enforcing supervisory control of nondeterministic discrete event systems with nondeterministic specifications. *Automatica*, 108:108470, 2019. doi:10.1016/j.automatica.2019.06.022.

A. F. Vaz and W. M. Wonham. On supervisor reduction in discrete-event systems. *International Journal of Control*, 44(2):475–491, 1986. doi:10.1080/00207178608933613.

W. M. Wonham and P. J. Ramadge. On the supremal controllable sublanguage of a given language. *SIAM Journal on Control and Optimization*, 25(3):637–659, 1987. doi:10.1137/0325036.